Annotated Z Bibliography

Jonathan Bowen
Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford OX1 3QD, UK.

Susan Stepney, Rosalind Barden
Logica Cambridge Limited,
Betjeman House, 104 Hills Road, Cambridge, CB2 1LQ, UK.

1 Introduction

This annotated Z bibliography contains a selected list of some pertinent publications for Z users. Most of those included are readily available, either as books or in journals. A few unpublished items have been included, where they are particularly relevant and can be obtained reasonably easily.

Some references are accompanied by an annotation. This may include a contents list (of a book), a list of the titles of Z related papers (in a collection) with cross references to the full details, or a summary of the work.

2 Cross references

The bibliography in the last section lists all references in alphabetical order by author. In this section papers are arranged by subject (with authors and brief details of the subject matter), together with cross references to the full details in the bibliography.

2.1 Management, style, and method

For justifications for using formality, and quick introductions to Z, see:

[63, 199] Cohen/McDermid. Justification of formal methods and notations
[204] Meyer. On formalism in specifications
[269] Spivey. Introduction to Z
[305, 306] Wing. General introduction to formal methods including Z
[311] Woodcock. Structuring specifications

For discussion about using formal methods in practice, see:

[20] Barden et al. Z in practice

[22, 45, 46, 118, 201] Barroca/McDermid, Bowen/Stavridou and Gerhart et al. Formal methods and safety-critical systems

[116] Gerhart. Applications of formal methods

[124, 40, 41, 42] Hall and Bowen/Hinchey. Myths and guidance about formal methods

[319] Worden. Fermenting and distilling ideas

Educational issues are presented and discussed in:

[68] Cooper. Educating management

[113] Garlan. Effective integration of formal methods into a professional master of software engineering course

[252] Saiedian. The mathematics of computing

[290] Swatman. Educating information systems professionals

Various papers describing good specification style are:

[6] Ainsworth et al. The use of viewpoint specifications, a technique with concentrates on making large specifications more understandable

[92] Duke. Enhancing structure

[122, 123] Gravell. Minimization in specification/design and what makes a good specification

[189] Macdonald. Usage and abusage

Much work has been done on attempting to integrate Z with traditional structured analysis methods. Some of this is described in:

[14] Aujla et al. A rigorous review technique

[55] Bryant et al. Structured methodologies and formal notations

[90] Draper. Z and SSADM

[83] Giovanni and Iachini. HOOD and Z

[239, 240, 238] Polack et al. SAZ Method – Structured Analysis and Z

[165, 166] Josephs and Redmond-Pyle. Entity-relationship models, structured methods, and Z

[245, 246] Randell. Data Flow Diagrams and Z

[256] Semmens and Allen. Yourdon and Z

[257] Semmens et al. Integrated structured analysis and formal specification techniques

[299] van Hee et al. Petri nets and Z

Other work towards the development of a ‘method’ for Z itself include:

[20] Barden et al. Z in practice

[127] Hall and McDermid. Towards a Z method using axiomatic specification in Z (using order sorted algebra and OBJ3 in particular)

[219] Neilson. A rigorous development method from Z to C
[308] Wood. A practical approach using Z and the refinement calculus
[321] Wordsworth. Software development with Z

The application of metrics to formal specifications has been studied:

[302, 17] Whitty and Bainbridge et al. Structural metrics for Z specifications

A formal specification in Z can be useful for deciding test cases, etc. Work on testing is reported in:

[8, 9] Ammann and Offutt. Functional and test specifications based on the category-partition method
[60] Carrington and Stocks. Formal methods and software testing
[77] Cusack and Wezeman. Deriving tests for objects specified in Z
[128] Hall. Testing with respect to formal specification
[133] Hayes. Specification directed module testing
[281] Stocks. Applying formal methods to software testing
[282] Stocks and Carrington. Deriving software test cases from formal specifications
[283, 284] Stocks and Carrington. Test templates: a specification-based testing framework and case study

2.2 Application areas

Surveys of formal methods, including Z users, are reported in:

[15] Austin and Parkin. Formal methods: a survey
[19] Barden et al. Use of Z (in the UK)
[71, 72, 73, 117, 118] Craigen et al. An international survey of major industrial formal methods applications, including a number using Z

One of the high profile users of Z is IBM UK Laboratories at Hursley for specification and development of the CICS transaction processing system. General descriptions of the CICS project include:

[64] Collins et al. Introducing formal methods: the CICS experience with Z
[147] Houston and King. CICS project report
[228] Nix and Collins. Use of software engineering and Z in the development of CICS
[235] Phillips. CICS experience throughout the lifecycle
[320] Wordsworth. The CICS Application Programming Interface (API) definition

Specifying secure systems is discussed in:

[160] Jones. Verification of critical properties
[267] Smith and Keighley. A secure transaction mechanism (SWORD secure DBMS)

Not all Z specifications are of software systems. Much interesting and important work has been done on formally specifying hardware, including microprocessors. The Inmos T800 transputer Floating Point Unit microcode development is a major real example where formal methods have saved time by reducing the amount of testing needed.

[196, 198, 197, 261, 260] May, Shepherd et al. T800 transputer FPU development

More technical papers on hardware applications (including embedded software) are:

[21] Barrett. A floating-point number system (IEEE standard)
[29, 30, 33] Bowen. Microprocessor instruction sets (Motorola M6800 and transputer)
[81, 82, 114, 135] Delisle/Garlan and Hayes. Oscilloscopes, including reuse of specifications
[167, 168] Kemp. Viper microprocessor
[266] Smith and Duke. Cache coherence protocol (in Object-Z)
[271] Spivey. Real-time kernel

Communications systems and protocols are specified in:

[29] Bowen et al. Network services via remote procedure calls (RPC)
[57] Butler. Service extension (PABX)
[95] Duke et al. Protocol specification and verification using Z
[98] Duke et al. Object-oriented protocol specification (mobile phone system, in Object-Z)
[121] Gotzhein. Open distributed systems
[132] Haughton. Safety and liveness properties of communication protocols
[194] Mataga and Zave. Formal specification of telephone features
[236] Pilling et al. Inheritance protocols for real-time scheduling
[293] Till and Potter. Gateway functions within a communications network
[323] Zave and Jackson. Specification of switching systems (PBX)

The following papers describe the use of Z for various graphics applications, standards (especially GKS), and human computer interfaces:

[2] Abowd et al. A survey of user interface languages including Z
[10, 11] Arnold et al. Configurable models of graphics systems (GKS)
[31, 34] Bowen. Formal specification of window systems (X in particular)
[53] Brown and Bowen. An extensible input system for Unix
[88] Dix et al. Human-Computer Interaction (HCI)
[91] Duce et al. Formal specification of *Presentation Environments for Multimedia Objects* (PREMO)
[130] Harrison. Engineering human-error tolerant software
[156] Johnson. Specification and prototyping of concurrent multi-user interfaces
[157] Johnson and Harrison. Declarative graphics and dynamic interaction
[214, 215] Narayana and Dharap. Formal specification of a Look Manager and a dialog system

[217] Nehlig and Duce. Formal specification of the GKS design primitive
[287] Sufrin. Formal specification of a display-oriented editor
[286, 289] Sufrin and He. Effective user interfaces and interactive processes
[294] Took. A formal design for an autonomous display manager

An important application area for formal methods is safety-critical systems where human lives may depend on correctness of the system.

[22, 45, 46, 118, 201] Barroca/McDermid, Bowen/Stavridou and Gerhart et al. Surveys covering formal methods and safety-critical systems
[35, 46, 39] Bowen et al. Safety-critical systems and standards
[173] Knight and Littlewood. Special issue of *IEEE Software* on *Safety-Critical Systems*
[237] Place and Kang. Safety-critical software: status report and annotated bibliography

Some examples of the application of Z to safety-critical systems are:

[152, 153, 154] Jacky. Formal specifications for a clinical cyclotron
[172] Knight and Kienzle. Using Z to specify a safety-critical system in the medical sector
[250] Ruddle. Specification of real-time safety-critical control systems

Other papers describing a variety of applications using Z include:

[1] Abowd et al. Software architectures
[32] Bowen. A text formatting tool
[36, 178] Bowen, Lano and Breuer. Reverse engineering
[54] Brownbridge. CASE toolset (for SSADM)
[56] Butcher. A behavioural semantics for Linda-2
[70] Craig. Specification of advanced AI architectures
[78, 79] de Barros and Harper. Formal specification and derivation of relational database applications
[104] Fenton and Mole. Flowgraph transformation
[209] Morgan and Sufrin. Specification of the Unix filing system
[216] Nash. Large systems
[248] Reizer et al. Requirements specification of a proposed POSIX standard
[276] Stepney. High integrity compilation
[288] Sufrin. A Z model of the Unix `make` utility
[315] Woodcock et al. Formal specification of the UK Defence Standard 00-56
[325] Zhang and Hitchcock. Designing knowledge-based systems and information systems

2.3 Textbooks on Z

[86] Diller. Z: an introduction to formal methods (2nd edition)

[139] Hayes et al. Specification case studies (the first book on Z, now in its 2nd edition, containing an excellent selection of example Z specifications)
[149] Imperato. An introduction to Z
[150] Ince. An introduction to discrete mathematics and formal system specification (2nd edition)
[183] Lightfoot. Formal specification using Z
[202] McMorran and Powell. Z guide for beginners.
[229] Norcliffe and Slater. Mathematics of software construction
[241] Potter, Sinclair and Till. An introduction to formal specification and Z (a popular first textbook on Z)
[247] Ratcliff. Introducing specification using Z
[317] Woodcock and Loomes. Software engineering mathematics
[321] Wordsworth. Software development with Z

A video course is also available [230, 231].

2.4 Language details

Z's syntax, semantics and mathematical toolkit are being internationally standardized under ISO/IEC JTC1/SC22. A draft version of the standard is available:

[52] Brien and Nicholls. Z Base Standard, version 1.0

The definition of the Z syntax and mathematical toolkit used by many practitioners is:

[273] Spivey. Z reference manual (2nd edition)

More technical works describing Z's formal semantics are:

[298] Diepen and van Hee. The link between Z and the relational algebra
[112] Gardiner et al. A simpler semantics
[268] Spivey. Understanding Z
[274] Spivey and Sufrin. Type inference

Z is often compared and contrasted with VDM (Vienna Development Method). The following papers show the cross-fertilization and comparisons between the two:

[26] Bera. Structuring for the VDM specification language, in response to the Z schema notation
[119] Gilmore. Correctness-oriented approaches to software development in which the Z, VDM and algebraic styles are compared
[138] Hayes. A comparative case study of VDM and Z
[141] Hayes et al. Understanding the differences between VDM and Z
[184, 185] Lindsay. A VDM perspective on reasoning about Z specifications and transferring VDM verification techniques to Z

[186] Lindsay and van Keulen. Case studies in the verification of specifications in VDM and Z
[206] Monahan and Shaw. Model-based specifications, including a discussion of the respective trade-offs in specification between Z and VDM

Reasoning about Z specifications is addressed in:

[208] Morgan and Sanders. Laws of the Logical Calculi
[309] Woodcock. Calculating properties (preconditions)
[314, 193] Woodcock/Brien and Martin. $\mathcal{W}$, a logic for Z.

Work on refining Z-like specifications towards an implementation (see also section 2.5) includes:

[21] Barrett. Refinement from Z to microcode via Occam
[16] Bailes and Duke. Class refinement
[23] Baumann. Z and natural semantics programming language theory for algorithm refinement
[85, 86] Diller. Hoare logic and Part II: *Methods of Reasoning*
[107, 108, 109] Fidge. Real-time refinement and program development
[119] Gilmore. Correctness-oriented approaches to software development (Z, VDM and algebraic styles are compared)
[144] He et al. Foundations for data refinement
[155] Jacob. Varieties of refinement
[163] Josephs. Data refinement calculation for Z specifications
[171] King and Sørensen. Specification and design of a library system
[177, 180] Lano and Haughton. Reasoning and refinement in object-oriented specification languages
[190, 191, 192] Mahoney/Hayes et al. Timed refinement
[218, 219] Neilson. Hierarchical refinement of Z specifications and a rigorous development method from Z to C
[258] Sennett. Using refinement to convince (pattern matching in ML)
[259] Sennett. Demonstrating the compliance of Ada programs with Z specifications
[289] Sufrin and He. Specification, analysis and refinement of interactive processes
[304] Whysall and McDermid. Object-oriented specification and refinement
[307] Wood. Software refinery
[312] Woodcock. Implementing promoted operations in Z
[318] Woodcock and Morgan. Refinement of state-based concurrent systems
[321] Wordsworth. Software development with Z

The 'refinement calculus' approach to refinement is espoused in:

[170] King. Z and the refinement calculus
[207] Morgan. A standard student textbook (2nd edition)
[210] Morgan and Vickers. Collected research papers
[308] Wood. A practical approach using Z and the refinement calculus

The related B-Method, with associated B-Tool, B-Toolkit and Abstract Machine Notation (AMN), have been developed by Abrial et al., also the progenator of Z:

[3, 4, 5] Abrial. The B-Tool, B-Method and forthcoming B-Book
[80] Dehbonei and Mejia. Use of B in the railways signalling industry
[87] Diller and Docherty. A comparison of Z and Abstract Machine Notation
[220, 221] Neilson and Prasad. ZedB (a prototype B-based proof tool)
[249] Ritchie et al. Experiences in using the Abstract Machine Notation in a GKS graphics standard case study
[285] Storey and Haughton. A strategy for the production of verifiable code using the B-Method

Execution of formal specifications is a subject of perennial debate. See:

[140] Hayes and Jones. Specifications are not (necessarily) executable

A retort may be found in:

[111] Fuchs. Specifications are (preferably) executable

Animating Z specifications is discussed in:

[51] Breuer and Bowen. Correct executable semantics for Z using abstract interpretation, including an informal taxonomy of approaches
[84] Dick et al. Computer aided transformation of Z into Prolog
[86] Diller. Part IV: *Specification Animation* (using Miranda)
[89] Doma and Nicholl. EZ: automatic prototyping
[120] Goodman. Animating Z specifications in Haskell using a monad
[131] Hasselbring. Animation of Object-Z specifications with a set-oriented prototyping language
[158] Johnson and Sanders. Functional implementations (Z to Miranda)
[187] Love. Animating Z specifications in SQL
[280] Stepney and Lord. An access control system (Z to Prolog)
[296] Valentine. Z^{--}, an executable subset of Z
[300] West and Eaglestone. Two approaches to animation (Z to Prolog)

Specific language features are addressed in:

[13, 263] Arthan and Smith. Free types in Z (including recursion)
[134] Hayes. A generalization of bags
[136] Hayes. Interpretations of schema operators
[137] Hayes. Multi-relations in Z (a cross between multi-sets and binary relations)
[188] Lupton. Promotion and forward simulation
[209] Morgan and Sufrin. Schema framing
[310, 312] Woodcock. Proof rules for promotion and implementing promoted operations

Some research has been undertaken in using and adapting Z to model concurrent systems:

- [66] Coombes and McDermid. Specifying distributed real-time systems
- [101, 102] Evans. Visualising, specifying and verifying concurrent systems using Z
- [156] Johnson. Applying temporal logic to support the specification and prototyping of concurrent multi-user interfaces
- [175] Lamport. TLZ: Temporal Logic of Actions (TLA) and Z
- [215] Narayana and Dharap. Invariant properties in a dialog system
- [255] Schuman et al. Object-oriented process specification

In particular, there has been some work on combining Z and CSP (Communicating Sequential Processes), a formal process model with associated algebra for concurrent systems:

- [24] Benjamin. A message passing system: an example of combining CSP and Z
- [164] Josephs. Theoretical work on a state-based approach to communicating processes
- [318] Woodcock and Morgan. Refinement of state-based concurrent systems

Researchers have also considered modelling and reasoning about real-time systems, for example, by combining temporal logic with Z.

- [66] Coombes and McDermid. Specifying temporal requirements for distributed real-time systems
- [99] Duke and Smith. Temporal logic and Z specifications
- [100] Engel. Specifying real-time systems with Z and the Duration Calculus
- [105] Fergus and Ince. Model logic and Z specifications item[[106]] Fidge. Specification and verification of real-time behaviour using Z and RTL
- [107, 108, 109] Fidge. Real-time refinement and program development
- [143] He Jifeng et al. Provably correct systems, including the use of Duration Calculus with schemas for structuring
- [156] Johnson. Applying temporal logic to support the specification and prototyping of concurrent multi-user interfaces
- [175] Lamport. TLZ: Temporal Logic of Actions (TLA) and Z
- [190, 191, 192] Mahoney/Hayes et al. Timed refinement
- [215] Narayana and Dharap. Invariant properties in a dialog system using Z and temporal logic
- [236] Pilling et al. Inheritance protocols for real-time scheduling
- [250] Ruddle. Specification of real-time safety-critical control systems
- [264] Smith. An object-oriented approach including a formalization of temporal logic history invariants

2.5 Conferences proceedings

Regular Z User Meetings are organized by the Z User Group and have had published proceedings since the 4th meeting:

[222] Nicholls. 4th Z User Meeting, Oxford, 1989
[224] Nicholls. 5th Z User Meeting, Oxford, 1990
[226] Nicholls. 6th Z User Meeting, York, 1991
[44] Bowen and Nicholls. 7th Z User Meeting, London, 1992
[38] Bowen and Hall. 8th Z User Meeting, Cambridge, 1994
[43] Bowen and Hinchey. 9th Z User Meeting, Limerick, 1995

The annual Refinement Workshop is organized by BCS-FACS. Papers cover a variety of refinement techniques from specification to code, and include some Z examples.

[200] McDermid. 1st Refinement Workshop, York, 1988
[211] Morgan and Woodcock. 3rd Refinement Workshop, Hursley, 1990
[212] Morris and Shaw. 4th Refinement Workshop, Cambridge, 1991
[160] Jones et al. 5th refinement Workshop, London, 1992
[292] Till. 6th refinement Workshop, London, 1994

FME Symposia are held every 18 months, organized by Formal Methods Europe. These grew out of the the later VDM-Europe conferences which included papers on Z:

[28] Bloomfield et al. VDM'88, Dublin
[27] Bjørner et al. VDM'90, Kiel
[242, 243] Prehn and Toetenel. VDM'91, Noordwijkerhout
[316] Woodcock and Larsen. FME'93, Odense
[213] Naftalin et al. FME'94, Barcelona

2.6 Tools

The ZIP Project tools catalogue lists some tools that support formatting, checking and proof of Z specifications:

[234] Parker. Z tools catalogue

Details of individual tools may be found in:

[12] Arthan. A proof tool based on HOL which grew into ProofPower (see below)
[37] Bowen and Gordon. Z and HOL (a tool based on higher order logic)
[110] Flynn et al. Formaliser (editor and type-checker)
[161] Jones. ICL ProofPower (a commercial tool based on HOL)
[162, 295] Jordan et al. CADiZ (formatter and type-checker)
[220, 221] Neilson and Prasad. ZedB (a prototype B-based schema expansion and precondition calculator tool)

[251] Saaltink. Z and EVES (a tool based on ZF set theory)
[272] Spivey. *f*UZZ (a commercial LaTeX formatter and type-checker, 2nd edition)
[322] Xiaoping Jia. ZTC (a freely available type-checker)

2.7 Object-Oriented Approaches

There has been much work recently to enhance Z with some of the structuring ideas from object-orientation. Overviews and comparisons can be found in:

[58] Carrington. ZOOM workshop report
[181] Lano and Haughton. Object-oriented specification case studies, many using extensions to Z
[278, 279] Stepney et al. Collected papers and a survey on object-orientation in Z

Object-Z is the best-documented and probably most widely used object-oriented extension to Z. The definitive description of the language is:

[97] Duke et al. Version 1 of Object-Z

Other Object-Z papers include:

[59] Carrington et al. Object-Z: an object-oriented extension to Z
[91] Duce et al. Formal specification of *Presentation Environments for Multimedia Objects* (PREMO)
[93] Duke and Duke. Towards a semantics
[94] Duke and Duke. Aspects of object-oriented specification (card game example)
[98] Duke et al. Object-oriented protocol specification (mobile phone system)
[131] Hasselbring. Animation with a set-oriented prototyping language
[244] Rafsanjani and Colwill. From Object-Z to C++
[266] Smith and Duke. Cache coherence protocol

Descriptions of other object-oriented approaches in conjunction with Z may be found in:

[7] Alencar and Goguen. OOZE: an object-oriented Z environment
[62] Chan and Trinder. An object-oriented data model supporting multi-methods, multiple inheritance, and static type-checking
[75] Cusack. Inheritance in object-oriented Z
[125, 126] Hall. A specification calculus for object-oriented systems and class hierarchies in Z
[129] Hammond. Producing Z specifications from object-oriented analysis
[176, 179, 36] Lano/Haughton et al. Z^{++}: an object-orientated extension to Z
[195] Maung and Howse. Hyper-Z: a new approach to object-orientation
[203] Meira and Cavalcanti. MooZ: Modular object-oriented Z specifications
[254, 255] Schuman, Pitt et al. Object-oriented subsystem and process specification
[301] Wezeman and Judge. Z for managed objects
[303, 304] Whysall and McDermid. Object-oriented specification and refinement

3 On-line information

The BIBTEX source for this bibliography and related information is available on-line via the World-Wide Web under the following URL (Uniform Resource Locator):

`http://www.comlab.ox.ac.uk/archive/z/bib.html`

The bibliography is searchable. The user may provide a regular expression or select from a number of predefined keywords. Hyperlinks are included to documents that can be accessed on-line.

4 Acknowledgements

We would like to thank all those who suggested references for inclusion in this bibliography. It has been adapted from the ZIP project bibliography [277], the on-line Z bibliography held at the Oxford University Computing Laboratory [47], including more recent additions.

Jonathan Bowen is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) grant no. GR/J15186.

5 Bibliography

[1] G. D. Abowd, R. Allen, and D. Garlan. Using style to understand descriptions of software architectures. *ACM Software Engineering Notes*, 18(5):9–20, December 1993.

[2] G. D. Abowd, J. P. Bowen, A. J. Dix, M. D. Harrison, and R. Took. User interface languages: A survey of existing methods. Technical Report PRG-TR-5-89, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, October 1989.

[3] J-R. Abrial. The B tool. In Bloomfield et al. [28], pages 86–87.

[4] J-R. Abrial. The B method for large software, specification, design and coding (abstract). In Prehn and Toetenel [243], pages 398–405. Volume 2: Tutorials.

[5] J.-R. Abrial. *The B-Book*. Cambridge University Press, 1995. To appear.

Contents: Mathematical reasoning; Set notation; Mathematical objects; Introduction to abstract machines; Formal definition of abstract machines; Theory of abstract machines; Construction large abstract machines; Example of abstract machines; Sequencing and loop; Programming examples; Refinement; Construction large software systems; Example of refinement;

Appendices: Summary of the most current notations; Syntax; Definitions; Visibility rules; Rules and axioms; Proof obligations.

[6] M. Ainsworth, A. H. Cruikchank, P. J. L. Wallis, and L. J. Groves. Viewpoint specification and Z. *Information and Software Technology*, 36(1):43–51, 1994.

[7] A. J. Alencar and J. A. Goguen. OOZE: An object-oriented Z environment. In P. America, editor, *Proc. ECOOP'91 European Conference on Object-Oriented Programming*, volume 512 of *Lecture Notes in Computer Science*, pages 180–199. Springer-Verlag, 1991.

[8] P. Ammann and J. Offutt. Functional and test specifications for the Mistix file system. Technical Report ISSE-TR-93-100, Department of Information & Software Systems Engineering, George Mason University, USA, January 1993.

[9] P. Ammann and J. Offutt. Using formal methods to mechanize category-partition testing. Technical Report ISSE-TR-93-105, Department of Information & Software Systems Engineering, George Mason University, USA, September 1993.

[10] D. B. Arnold, D. A. Duce, and G. J. Reynolds. An approach to the formal specification of configurable models of graphics systems. In G. Maréchal, editor, *Proc. EUROGRAPHICS'87, European Computer Graphics Conference and Exhibition*, pages 439–463. Elsevier Science Publishers (North-Holland), 1987.

The paper describes a general framework for the formal specification of modular graphics systems, illustrated by an example taken from the Graphical Kernel System (GKS) standard.

[11] D. B. Arnold and G. J. Reynolds. Configuring graphics systems components. *IEE/BCS Software Engineering Journal*, 3(6):248–256, November 1988.

[12] R. D. Arthan. Formal specification of a proof tool. In Prehn and Toetenel [242], pages 356–370. Volume 1: Conference Contributions.

[13] R. D. Arthan. On free type definitions in Z. In Nicholls [226], pages 40–58.

[14] S. Aujla, A. Bryant, and L. Semmens. A rigorous review technique: Using formal notations within conventional development methods. In *Proc. 1993 Software Engineering Standards Symposium*, pages 247–255. IEEE Computer Society Press, 1993.

[15] S. Austin and G. I. Parkin. Formal methods: A survey. Technical report, National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW, UK, March 1993.

[16] C. Bailes and R. Duke. The ecology of class refinement. In Morris and Shaw [212], pages 185–196.

[17] J. Bainbridge, R. W. Whitty, and J. B. Wordsworth. Obtaining structural metrics of Z specifications for systems development. In Nicholls [224], pages 269–281.

[18] R. Barden and S. Stepney. Support for using Z. In Bowen and Nicholls [44], pages 255–280.

[19] R. Barden, S. Stepney, and D. Cooper. The use of Z. In Nicholls [226], pages 99–124.

[20] R. Barden, S. Stepney, and D. Cooper. *Z in Practice*. BCS Practitioner Series. Prentice Hall, 1994.

[21] G. Barrett. Formal methods applied to a floating-point number system. *IEEE Transactions on Software Engineering*, 15(5):611–621, May 1989.

This paper presents a formalization of the IEEE standard for binary floating-point arithmetic in Z. The formal specification is refined into four components. The procedures presented form the basis for the floating-point unit of the Inmos IMS T800 transputer. This work resulted in a joint UK Queen's Award for Technological Achievement for Inmos Ltd and the Oxford University Computing Laboratory in 1990. It was estimated that the approach saved a year in development time compared to traditional methods.

[22] L. M. Barroca and J. A. McDermid. Formal methods: Use and relevance for the development of safety-critical systems. *The Computer Journal*, 35(6):579–599, December 1992.

[23] P. Baumann. Z and natural semantics. In Bowen and Hall [38], pages 168–184.

[24] M. Benjamin. A message passing system: An example of combining CSP and Z. In Nicholls [222], pages 221–228.

[25] M. Benveniste. Writing operational semantics in Z: A structural approach. In Prehn and Toetenel [242], pages 164–188. Volume 1: Conference Contributions.

[26] S. Bera. Structuring for the VDM specification language. In Bloomfield et al. [28], pages 2–25.

[27] D. Bjørner, C. A. R. Hoare, and H. Langmaack, editors. *VDM and Z – Formal Methods in Software Development*, volume 428 of *Lecture Notes in Computer Science*. VDM-Europe, Springer-Verlag, 1990.

The 3rd VDM-Europe Symposium was held at Kiel, Germany, 17–21 April 1990. A significant number of papers concerned with Z were presented [61, 93, 114, 83, 121, 125, 170, 253, 275, 298, 318].

[28] R. Bloomfield, L. Marshall, and R. Jones, editors. *VDM – The Way Ahead*, volume 328 of *Lecture Notes in Computer Science*. VDM-Europe, Springer-Verlag, 1988.

The 2nd VDM-Europe Symposium was held at Dublin, Ireland, 11–16 September 1988. See [3, 26].

[29] J. P. Bowen. Formal specification and documentation of microprocessor instruction sets. *Microprocessing and Microprogramming*, 21(1–5):223–230, August 1987.

[30] J. P. Bowen. The formal specification of a microprocessor instruction set. Technical Monograph PRG-60, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, January 1987.

The Z notation is used to define the Motorola M6800 8-bit microprocessor instruction set.

[31] J. P. Bowen. Formal specification of window systems. Technical Monograph PRG-74, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, June 1989.

Three existing window systems, X from MIT, WM from Carnegie-Mellon University and the Blit from AT&T Bell Laboratories are covered.

[32] J. P. Bowen. POS: Formal specification of a UNIX tool. *IEE/BCS Software Engineering Journal*, 4(1):67–72, January 1989.

[33] J. P. Bowen. Formal specification of the ProCoS/safemos instruction set. *Microprocessors and Microsystems*, 14(10):631–643, December 1990.

This article is part of a special issue on *Formal aspects of microprocessor design*, edited by H. S. M. Zedan. See also [260].

[34] J. P. Bowen. X: Why Z? *Computer Graphics Forum*, 11(4):221–234, October 1992.

This paper asks whether window management systems would not be better specified through a formal methodology and gives examples in Z of X11.

[35] J. P. Bowen. Formal methods in safety-critical standards. In *Proc. 1993 Software Engineering Standards Symposium*, pages 168–177. IEEE Computer Society Press, 1993.

[36] J. P. Bowen, P. T. Breuer, and K. C. Lano. Formal specifications in software maintenance: From code to Z^{++} and back again. *Information and Software Technology*, 35(11/12):679–690, November/December 1993.

[37] J. P. Bowen and M. J. C. Gordon. Z and HOL. In Bowen and Hall [38], pages 141–167.

[38] J. P. Bowen and J. A. Hall, editors. *Z User Workshop, Cambridge 1994*, Workshops in Computing. Springer-Verlag, 1994.

Proceedings of the Eigth Annual Z User Meeting, St. John's College, Cambridge, UK. Published in collaboration with the British Computer Society. For individual papers, see [23, 37, 48, 47, 51, 60, 62, 87, 100, 102, 113, 126, 127, 129, 131, 175, 194, 238, 265, 301, 315, 319]. The proceedings also includes an *Introduction and Opening Remarks*, a *Select Z Bibliography* [48] and a section answering *Frequently Asked Questions* [47].

[39] J. P. Bowen and M. G. Hinchey. Formal methods and safety-critical standards. *IEEE Computer*, 27(8):68–71, August 1994.

[40] J. P. Bowen and M. G. Hinchey. Seven more myths of formal methods. Technical Report PRG-TR-7-94, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, June 1994. Revised version in *IEEE Software*.

This article deals with further myths in addition to those presented in [124].

[41] J. P. Bowen and M. G. Hinchey. Seven more myths of formal methods: Dispelling industrial prejudices. In Naftalin et al. [213], pages 105–117.

[42] J. P. Bowen and M. G. Hinchey. Ten commandments of formal methods. Technical Report 350, University of Cambridge, Computer Laboratory, UK, September 1994. Revised version in *IEEE Computer*.

[43] J. P. Bowen and M. G. Hinchey, editors. *Z User Workshop, Limerick 1995*, Workshops in Computing. Springer-Verlag, 1995. To appear.

[44] J. P. Bowen and J. E. Nicholls, editors. *Z User Workshop, London 1992*, Workshops in Computing. Springer-Verlag, 1993.

Proceedings of the Seventh Annual Z User Meeting, DTI Offices, London, UK. Published in collaboration with the British Computer Society. For individual papers, see [18, 49, 67, 71, 77, 76, 90, 142, 153, 172, 182, 187, 195, 227, 232, 244, 250, 290, 297]. The proceedings also includes an *Introduction and Opening Remarks*, a *Select Z Bibliography* and a section answering *Frequently Asked Questions*.

[45] J. P. Bowen and V. Stavridou. The industrial take-up of formal methods in safety-critical and other areas: A perspective. In Woodcock and Larsen [316], pages 183–195.

[46] J. P. Bowen and V. Stavridou. Safety-critical systems, formal methods and standards. *IEE/BCS Software Engineering Journal*, 8(4):189–209, July 1993.

A survey on the use of formal methods, including B and Z, for safety-critical systems. A previous version is also available as Oxford University Computing Laboratory Technical Report PRG-TR-5-92.

[47] J.P. Bowen. Comp.specification.z and Z FORUM frequently asked questions. In Bowen and Hall [38], pages 397–404.

[48] J.P. Bowen. Select Z bibliography. In Bowen and Hall [38], pages 359–396.

[49] A. Bradley. Requirements for Defence Standard 00-55. In Bowen and Nicholls [44], pages 93–94.

[50] P. T. Breuer. Z! in progress: Maintaining Z specifications. In Nicholls [224], pages 295–318.

[51] P. T. Breuer and J. P. Bowen. Towards correct executable semantics for Z. In Bowen and Hall [38], pages 185–209.

[52] S. M. Brien and J. E. Nicholls. Z base standard. Technical Monograph PRG-107, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, November 1992. Accepted for standardization under ISO/IEC JTC1/SC22.

This is the first publicly available version of the proposed ISO Z Standard. See also [273] for the current most widely available Z reference manual.

[53] D. J. Brown and J. P. Bowen. The Event Queue: An extensible input system for UNIX workstations. In *Proc. European Unix Users Group Conference*, pages 29–52. EUUG, May 1987.

Available from EUUG Secretariat, Owles Hall, Buntingford, Hertfordshire SG9 9PL, UK.

[54] D. Brownbridge. Using Z to develop a CASE toolset. In Nicholls [222], pages 142–149.

[55] A. Bryant. Structured methodologies and formal notations: Developing a framework for synthesis and investigation. In Nicholls [222], pages 229–241.

[56] P. Butcher. A behavioural semantics for Linda-2. *IEE/BCS Software Engineering Journal*, 6(4):196–204, July 1991.

[57] M. J. Butler. Service extension at the specification level. In Nicholls [224], pages 319–336.

[58] D. Carrington. ZOOM workshop report. In Nicholls [226], pages 352–364.

This paper records the activities of a workshop on Z and object-oriented methods held in August 1992 at Oxford. A comprehensive bibliography is included.

[59] D. Carrington, D. Duke, R. Duke, P. King, G. A. Rose, and G. Smith. Object-Z: An object-oriented extension to Z. In S. Vuong, editor, *Formal Description Techniques, II (FORTE'89)*, pages 281–296. Elsevier Science Publishers (North-Holland), 1990.

[60] D. Carrington and P. Stocks. A tale of two paradigms: Formal methods and software testing. In Bowen and Hall [38], pages 51–68.

Also available as Technical Report 94-4, Department of Computer Science, University of Queensland, 1994.

[61] P. Chalin and P. Grogono. Z specification of an object manager. In Bjørner et al. [27], pages 41–71.

[62] D. K. C. Chan and P. W. Trinder. An object-oriented data model supporting multi-methods, multiple inheritance, and static type checking: A specification in Z. In Bowen and Hall [38], pages 297–315.

[63] B. Cohen. Justification of formal methods for system specifications & A rejustification of formal notations. *IEE/BCS Software Engineering Journal*, 4(1):26–38, January 1989.

[64] B. P. Collins, J. E. Nicholls, and I. H. Sørensen. Introducing formal methods: The CICS experience with Z. In B. Neumann et al., editors, *Mathematical Structures for Software Engineering*. Oxford University Press, 1991.

[65] A. Coombes and J. A. McDermid. A tool for defining the architecture of Z specifications. In Nicholls [224], pages 77–92.

[66] A. Coombes and J. A. McDermid. Specifying temporal requirements for distributed real-time systems in Z. Computer Science Report YCS176, University of York, Heslington, York YO1 5DD, UK, 1992.

[67] A. C. Coombes and J. A. McDermid. Using diagrams to give a formal specification of timing constraints in Z. In Bowen and Nicholls [44], pages 119–130.

[68] D. Cooper. Educating management in Z. In Nicholls [222], pages 192–194.

[69] V. A. O. Cordeiro, A. C. A. Sampaio, and S. L. Meira. From MooZ to Eiffel – a rigorous approach to system development. In Naftalin et al. [213], pages 306–325.

[70] I. Craig. *The Formal Specification of Advanced AI Architectures*. AI Series. Ellis Horwood, September 1991.

This book contains two rather large (and relatively complete) specifications of Artificial Intelligence (AI) systems using Z. The architectures are the blackboard and Cassandra architectures. As well as showing that formal specification *can* be used in AI at the architecture level, the book is intended as a case-studies book, and also contains introductory material on Z (for AI people). The book assumes a knowledge of Z, so for non-AI people its primary use is for the presentation of the large specifications. The blackboard specification, with explanatory text, is around 100 pages.

[71] D. Craigen, S. L. Gerhart, and T. Ralston. An international survey of industrial applications of formal methods. In Bowen and Nicholls [44], pages 1–5.

[72] D. Craigen, S. L. Gerhart, and T. J. Ralston. Formal methods reality check: Industrial usage. In Woodcock and Larsen [316], pages 250–267.

[73] D. Craigen, S. L. Gerhart, and T. J. Ralston. An international survey of industrial applications of formal methods. Technical Report NIST GCR 93/626-V1 & 2, Atomic Energy Control Board of Canada, US National Institute of Standards and Technology, and US Naval Research Laboratories, 1993.

Volume 1: Purpose, Approach, Analysis and Conclusions; Volume 2: Case Studies. Order numbers: PB93-178556/AS & PB93-178564/AS; National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161, USA.

[74] D. Craigen, S. Kromodimoeljo, I. Meisels, W. Pase, and M. Saaltink. EVES: An overview. In Prehn and Toetenel [242], pages 389–405. Volume 1: Conference Contributions.

[75] E. Cusack. Inheritance in object oriented Z. In P. America, editor, *Proc. ECOOP'91 European Conference on Object-Oriented Programming*, volume 512 of *Lecture Notes in Computer Science*, pages 167–179. Springer-Verlag, 1991.

[76] E. Cusack. Using Z in communications engineering. In Bowen and Nicholls [44], pages 196–202.

[77] E. Cusack and C. Wezeman. Deriving tests for objects specified in Z. In Bowen and Nicholls [44], pages 180–195.

[78] R. S. M. de Barros. Deriving relational database programs from formal specifications. In Naftalin et al. [213], pages 703–723.

[79] R. S. M. de Barros and D. J. Harper. A method for the specification of relational database applications. In Nicholls [226], pages 261–286.

[80] B. Dehbonei and F. Mejia. Formal methods in the railways signalling industry. In Naftalin et al. [213], pages 26–34.

[81] N. Delisle and D. Garlan. Formally specifying electronic instruments. In *Proc. Fifth International Workshop on Software Specification and Design*. IEEE Computer Society, May 1989. Also published in ACM SIGSOFT Software Engineering Notes 14(3).

[82] N. Delisle and D. Garlan. A formal specification of an oscilloscope. *IEEE Software*, 7(5):29–36, September 1990.

Unlike most work on the application of formal methods, this research uses formal methods to gain insight into system architecture. The context for this case study is electronic instrument design.

[83] R. Di Giovanni and P. L. Iachini. HOOD and Z for the development of complex systems. In Bjørner et al. [27], pages 262–289.

[84] A. J. J. Dick, P. J. Krause, and J. Cozens. Computer aided transformation of Z into Prolog. In Nicholls [222], pages 71–85.

[85] A. Diller. Z and Hoare logics. In Nicholls [226], pages 59–76.

[86] A. Diller. *Z: An Introduction to Formal Methods*. John Wiley & Sons, Chichester, UK, 2nd edition, 1994.

This book offers a comprehensive tutorial to Z from the practical viewpoint. Many natural deduction style proofs are presented and exercises are included. Z as defined in the 2nd edition of *The Z Notation* [273] is used throughout.

Contents: Tutorial introduction; Methods of reasoning; Case studies; Specification animation; Reference manual; Answers to exercises; Glossaries of terms and symbols; Bibliography.

[87] A. Diller and R. Docherty. Z and abstract machine notation: A comparison. In Bowen and Hall [38], pages 250–263.

[88] A. J. Dix, J. Finlay, G. D. Abowd, and R. Beale. *Human-Computer Interaction*. Prentice Hall International, 1993.

[89] V. Doma and R. Nicholl. EZ: A system for automatic prototyping of Z specifications. In Prehn and Toetenel [242], pages 189–203. Volume 1: Conference Contributions.

[90] C. Draper. Practical experiences of Z and SSADM. In Bowen and Nicholls [44], pages 240–251.

[91] D. A. Duce, D. J. Duke, P. J. W. ten Hagen, and G. J. Reynolds. PREMO - an initial approach to a formal definition. *Computer Graphics Forum*, 13(3):C–393–C–406, 1994.

PREMO (Presentation Environments for Multimedia Objects) is a work item proposal by the ISO/IEC JTC11/SC24 committee, which is responsible for international standardization in the area of computer graphics and image processing.

[92] D. Duke. Enhancing the structures of Z specifications. In Nicholls [226], pages 329–351.

[93] D. Duke and R. Duke. Towards a semantics for Object-Z. In Bjørner et al. [27], pages 244–261.

[94] R. Duke and D. Duke. Aspects of object-oriented formal specification. In *Proc. 5th Australian Software Engineering Conference (ASWEC'90)*, pages 21–26, 1990.

[95] R. Duke, I. J. Hayes, P. King, and G. A. Rose. Protocol specification and verification using Z. In S. Aggarwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification VIII*, pages 33–46. Elsevier Science Publishers (North-Holland), 1988.

[96] R. Duke, P. King, G. A. Rose, and G. Smith. The Object-Z specification language. In T. Korson, V. Vaishnavi, and B. Meyer, editors, *Technology of Object-Oriented Languages and Systems: TOOLS 5*, pages 465–483. Prentice Hall, 1991.

[97] R. Duke, P. King, G. A. Rose, and G. Smith. The Object-Z specification language: Version 1. Technical Report 91-1, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, April 1991.

The most complete (and currently the standard) reference on Object-Z. It has been reprinted by ISO JTC1 WG7 as document number 372. A condensed version of this report was published as [96].

[98] R. Duke, G. A. Rose, and A. Lee. Object-oriented protocol specification. In L. Logrippo, R. L. Probert, and H. Ural, editors, *Protocol Specification, Testing, and Verification X*, pages 325–338. Elsevier Science Publishers (North-Holland), 1990.

[99] R. Duke and G. Smith. Temporal logic and Z specifications. *Australian Computer Journal*, 21(2):62–69, May 1989.

[100] M. Engel. Specifying real-time systems with Z and the Duration Calculus. In Bowen and Hall [38], pages 282–294.

[101] A. S. Evans. Specifying & verifying concurrent systems using Z. In Naftalin et al. [213], pages 366–400.

[102] A. S. Evans. Visualising concurrent Z specifications. In Bowen and Hall [38], pages 269–281.

[103] P. C. Fencott, A. J. Galloway, M. A. Lockyer, S. J. O'Brien, and S. Pearson. Formalising the semantics of Ward/Mellor SA/RT essential models using a process algebra. In Naftalin et al. [213], pages 681–702.

[104] N. E. Fenton and D. Mole. A note on the use of Z for flowgraph transformation. *Information and Software Technology*, 30(7):432–437, 1988.

[105] E. Fergus and D. C. Ince. Z specifications and modal logic. In P. A. V. Hall, editor, *Proc. Software Engineering 90*, volume 1 of *British Computer Society Conference Series*. Cambridge University Press, 1990.

[106] C. J. Fidge. Specification and verification of real-time behaviour using Z and RTL. In J. Vytopil, editor, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Lecture Notes in Computer Science, pages 393–410. Springer-Verlag, 1992.

[107] C. J. Fidge. Real-time refinement. In Woodcock and Larsen [316], pages 314–331.

[108] C. J. Fidge. Adding real time to formal program development. In Naftalin et al. [213], pages 618–638.

[109] C. J. Fidge. Proof obligations for real-time refinement. In Till [292], pages 279–305.

[110] M. Flynn, T. Hoverd, and D. Brazier. Formaliser – an interactive support tool for Z. In Nicholls [222], pages 128–141.

[111] N. E. Fuchs. Specifications are (preferably) executable. *IEE/BCS Software Engineering Journal*, 7(5):323–334, September 1992.

[112] P. H. B. Gardiner, P. J. Lupton, and J. C. P. Woodcock. A simpler semantics for Z. In Nicholls [224], pages 3–11.

[113] D. Garlan. Integrating formal methods into a professional master of software engineering program. In Bowen and Hall [38], pages 71–85.

[114] D. Garlan and N. Delisle. Formal specifications as reusable frameworks. In Bjørner et al. [27], pages 150–163.

[115] D. Garlan and D. Notkin. Formalizing design spaces: Implicit invocation mechanisms. In Prehn and Toetenel [242], pages 31–45. Volume 1: Conference Contributions.

[116] S. L. Gerhart. Applications of formal methods: Developing virtuoso software. *IEEE Software*, 7(5):6–10, September 1990.

This is an introduction to a special issue on Formal Methods with an emphasis on Z in particular. It was published in conjunction with special Formal Methods issues of *IEEE Transactions on Software Engineering* and *IEEE Computer*. See also [82, 124, 214, 271, 305].

[117] S. L. Gerhart, D. Craigen, and T. Ralston. Observations on industrial practice using formal methods. In *Proc. 15th International Conference on Software Engineering (ICSE), Baltimore, Maryland, USA*, May 1993.

[118] S. L. Gerhart, D. Craigen, and T. Ralston. Experience with formal methods in critical systems. *IEEE Software*, 11(1):21–28, January 1994.

Several commercial and exploratory cases in which Z features heavily are briefly presented on page 24. See also [173].

[119] S. Gilmore. Correctness-oriented approaches to software development. Technical Report ECS-LFCS-91-147 (also CST-76-91), Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, UK, 1991.

This PhD thesis provides a critical evaluation of Z, VDM and algebraic specifications.

[120] H. S. Goodman. Animating Z specifications in Haskell using a monad. Technical Report CSR-93-10, School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK, November 1993.

[121] R. Gotzhein. Specifying open distributed systems with Z. In Bjørner et al. [27], pages 319–339.

[122] A. M. Gravell. Minimisation in formal specification and design. In Nicholls [222], pages 32–45.

[123] A. M. Gravell. What is a good formal specification? In Nicholls [224], pages 137–150.

[124] J. A. Hall. Seven myths of formal methods. *IEEE Software*, 7(5):11–19, September 1990.

Formal methods are difficult, expensive, and not widely useful, detractors say. Using a case study and other real-world examples, this article challenges such common myths.

[125] J. A. Hall. Using Z as a specification calculus for object-oriented systems. In Bjørner et al. [27], pages 290–318.

[126] J. A. Hall. Specifying and interpreting class hierarchies in Z. In Bowen and Hall [38], pages 120–138.

[127] J. G. Hall and J. A. McDermid. Towards a Z method: Axiomatic specification in Z. In Bowen and Hall [38], pages 213–229.

[128] P. A. V. Hall. Towards testing with respect to formal specification. In *Proc. Second IEE/BCS Conference on Software Engineering*, number 290 in Conference Publication, pages 159–163. IEE/BCS, July 1988.

[129] J. A. R. Hammond. Producing Z specifications from object-oriented analysis. In Bowen and Hall [38], pages 316–336.

[130] M. D. Harrison. Engineering human-error tolerant software. In Nicholls [226], pages 191–204.

[131] W. Hasselbring. Animation of Object-Z specifications with a set-oriented prototyping language. In Bowen and Hall [38], pages 337–356.

[132] H. P. Haughton. Using Z to model and analyse safety and liveness properties of communication protocols. *Information and Software Technology*, 33(8):575–580, October 1991.

[133] I. J. Hayes. Specification directed module testing. *IEEE Transactions on Software Engineering*, 12(1):124–133, January 1986.

[134] I. J. Hayes. A generalisation of bags in Z. In Nicholls [222], pages 113–127.

[135] I. J. Hayes. Specifying physical limitations: A case study of an oscilloscope. Technical Report 167, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, July 1990.

[136] I. J. Hayes. Interpretations of Z schema operators. In Nicholls [224], pages 12–26.

[137] I. J. Hayes. Multi-relations in Z: A cross between multi-sets and binary relations. *Acta Informatica*, 29(1):33–62, February 1992.

[138] I. J. Hayes. VDM and Z: A comparative case study. *Formal Aspects of Computing*, 4(1):76–99, 1992.

[139] I. J. Hayes, editor. *Specification Case Studies*. Prentice Hall International Series in Computer Science, 2nd edition, 1993.

This is a revised edition of the first ever book on Z, originally published in 1987; it contains substantial changes to every chapter. The notation has been revised to be consistent with *The Z Notation: A Reference Manual* by Mike Spivey [273]. The CAVIAR chapter has been extensively changed to make use of a form of modularization.

Divided into four sections, the first provides tutorial examples of specifications, the second is devoted to the area of software engineering, the third covers distributed computing, analyzing the role of mathematical specification, and the fourth part covers the IBM CICS transaction processing system. Appendices include comprehensive glossaries of the Z mathematical and schema notation. The book will be of interest to the professional software engineer involved in designing and specifying large software projects.

The other contributors are W. Flinn, R. B. Gimson, S. King, C. C. Morgan, I. H. Sørensen and B. A. Sufrin.

[140] I. J. Hayes and C. B. Jones. Specifications are not (necessarily) executable. *IEE/BCS Software Engineering Journal*, 4(6):330–338, November 1989.

[141] I. J. Hayes, C. B. Jones, and J. E. Nicholls. Understanding the differences between VDM and Z. *FACS Europe*, Series I, 1(1):7–30, Autumn 1993.

Also available as Technical Report UMCS-93-8-1, Department of Computer Science, University of Manchester, UK, 1993.

[142] I. J. Hayes and L. Wildman. Towards libraries for Z. In Bowen and Nicholls [44], pages 9–36.

[143] He Jifeng, C. A. R. Hoare, M. Fränzle, Markus Müller-Ulm, E.-R. Olderog, M. Schenke, A. P. Ravn, and H. Rischel. Provably correct systems. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real Time and Fault Tolerant Systems*, volume 863 of *Lecture Notes in Computer Science*, pages 288–335. Springer-Verlag, 1994.

[144] He Jifeng, C. A. R. Hoare, and J. W. Sanders. Data refinement refined. In B. Robinet and R. Wilhelm, editors, *Proc. ESOP 86*, volume 213 of *Lecture Notes in Computer Science*, pages 187–196. Springer-Verlag, 1986.

[145] B. Hepworth. ZIP: A unification initiative for Z standards, methods and tools. In Nicholls [222], pages 253–259.

[146] B. Hepworth and D. Simpson. The ZIP project. In Nicholls [224], pages 129–133.

[147] I. S. C. Houston and S. King. CICS project report: Experiences and results from the use of Z in IBM. In Prehn and Toetenel [242], pages 588–596. Volume 1: Conference Contributions.

[148] P. L. Iachini. Operation schema iterations. In Nicholls [224], pages 50–57.

[149] M. Imperato. *An Introduction to Z.* Chartwell-Bratt, 1991.

Contents: Introduction; Set theory; Logic; Building Z specifications; Relations; Functions; Sequences; Bags; Advanced Z; Case study: a simple banking system.

[150] D.C. Ince. *An Introduction to Discrete Mathematics, Formal System Specification and Z.* Oxford Applied Mathematics and Computing Science Series. Oxford University Press, 2nd edition, 1993.

[151] D. Jackson. Abstract model checking of infinite specifications. In Naftalin et al. [213], pages 519–531.

[152] J. Jacky. Formal specifications for a clinical cyclotron control system. *ACM SIGSOFT Software Engineering Notes*, 15(4):45–54, September 1990.

[153] J. Jacky. Formal specification and development of control system input/output. In Bowen and Nicholls [44], pages 95–108.

[154] J. Jacky. Specifying a safety-critical control system in Z. In Woodcock and Larsen [316], pages 388–402.

[155] J. Jacob. The varieties of refinements. In Morris and Shaw [212], pages 441–455.

[156] C. W. Johnson. Applying temporal logic to support the specification and prototyping of concurrent multi-user interfaces. In D. Diaper and N. Hammond, editors, *People and Computers VI: Usability Now*, pages 145–156. Cambridge University Press, 1991.

[157] C. W. Johnson and M. D. Harrison. Declarative graphics and dynamic interaction. In F. H. Post and W. Barth, editors, *Proc. EUROGRAPHICS'91*, pages 195–207. Elsevier Science Publications (North-Holland), 1991.

[158] M. Johnson and P. Sanders. From Z specifications to functional implementations. In Nicholls [222], pages 86–112.

[159] C. B. Jones. Interference revisited. In Nicholls [224], pages 58–73.

[160] C. B. Jones, R. C. Shaw, and T. Denvir, editors. *5th Refinement Workshop*, Workshop in Computing. BCS-FACS, Springer-Verlag, 1992.

The workshop was held at Lloyd's Register, London, UK, 8–10 January 1992. See [259].

[161] R. B. Jones. ICL ProofPower. *BCS FACS FACTS*, Series III, 1(1):10–13, Winter 1992.

[162] D. Jordan, J. A. McDermid, and I. Toyn. CADiZ – computer aided design in Z. In Nicholls [224], pages 93–104.

[163] M. B. Josephs. The data refinement calculator for Z specifications. *Information Processing Letters*, 27(1):29–33, 1988.

[164] M. B. Josephs. A state-based approach to communicating processes. *Distributed Computing*, 3:9–18, 1988.

A theoretical paper on combining features of CSP and Z.

[165] M. B. Josephs. Specifying reactive systems in Z. Technical Report PRG-TR-19-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, July 1991.

[166] M. B. Josephs and D. Redmond-Pyle. Entity-relationship models expressed in Z: A synthesis of structured and formal methods. Technical Report PRG-TR-20-91, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, July 1991.

[167] D. H. Kemp. Specification of Viper1 in Z. Memorandum no. 4195, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, October 1988.

[168] D. H. Kemp. Specification of Viper2 in Z. Memorandum no. 4217, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, October 1988.

[169] P. King. Printing Z and Object-Z LaTeX documents. Department of Computer Science, University of Queensland, May 1990.

A description of a Z style option '`oz.sty`', an extended version of Mike Spivey's '`zed.sty`' [270], for use with the LaTeX document preparation system [174]. It is particularly useful for printing Object-Z documents [59, 93].

[170] S. King. Z and the refinement calculus. In Bjørner et al. [27], pages 164–188.

Also published as Technical Monograph PRG-79, Oxford University Computing Laboratory, February 1990.

[171] S. King and I. H. Sørensen. Specification and design of a library system. In McDermid [200].

[172] J. C. Knight and D. M. Kienzle. Preliminary experience using Z to specify a safety-critical system. In Bowen and Nicholls [44], pages 109–118.

[173] J. C. Knight and B. Littlewood. Critical task of writing dependable software. *IEEE Software*, 11(1):16–20, January 1994.

Guest editors' introduction to a special issue of *IEEE Software* on *Safety-Critical Systems*. A short section on formal methods mentions several Z books on page 18. See also [118].

[174] L. Lamport. *LaTeX User's Guide & Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1986.

Z specifications may be produced using the document preparation system LaTeX together with a special LaTeX style option. The most widely used style files are `fuzz.sty` [272], `zed.sty` [270] and `oz.sty` [169].

[175] L. Lamport. TLZ. In Bowen and Hall [38], pages 267–268. Abstract.

[176] K. C. Lano. Z^{++}, an object-orientated extension to Z. In Nicholls [224], pages 151–172.

[177] K. C. Lano. Refinement in object-oriented specification languages. In Till [292], pages 236–259.

[178] K. C. Lano and P. T. Breuer. From programs to Z specifications. In Nicholls [222], pages 46–70.

[179] K. C. Lano and H. P. Haughton. An algebraic semantics for the specification language Z^{++}. In *Proc. Algebraic Methodology and Software Technology Conference (AMAST '91)*. Springer-Verlag, 1992.

[180] K. C. Lano and H. P. Haughton. Reasoning and refinement in object-oriented specification languages. In O. L. Madsen, editor, *ECOOP '92: European Conference on Object-Oriented Programming*, volume 615 of *Lecture Notes in Computer Science*, pages 78–97. Springer-Verlag, 1992.

[181] K. C. Lano and H. P. Haughton, editors. *Object Oriented Specification Case Studies*. Object Oriented Series. Prentice Hall International, 1993.

Contents: Chapters introducing object oriented methods, object oriented formal specification and the links between formal and structured object-oriented techniques; seven case studies in particular object oriented formal methods, including:

> The Unix Filing System: A MooZ Specification; An Object-Z Specification of a Mobile Phone System; Object-oriented Specification in VDM^{++}; Specifying a Concept-recognition System in Z^{++}; Specification in OOZE; Refinement in Fresco; SmallVDM: An Environment for Formal Specification and Prototyping in Smalltalk.

A glossary, index and bibliography are also included. The contributors are some of the leading figures in the area, including the developers of the above methods and languages: Silvio Meira, Gordon Rose, Roger Duke, Antonio Alencar, Joseph Goguen, Alan Wills, Cassio Souza dos Santos, Ana Cavalcanti.

[182] K. C. Lano and H. P. Haughton. Reuse and adaptation of Z specifications. In Bowen and Nicholls [44], pages 62–90.

[183] D. Lightfoot. *Formal Specification using* Z. Macmillan, 1991.

Contents: Introduction; Sets in Z; Using sets to describe a system – a simple example; Logic: propositional calculus; Example of a Z specification document; Logic: predicate calculus; Relations; Functions; A seat allocation system; Sequences; An example of sequences – the aircraft example again; Extending a specification; Collected notation; Books on formal specification; Hints on creating specifications; Solutions to exercises. Also available in French.

[184] P. A. Lindsay. Reasoning about Z specifications: A VDM perspective. Technical Report 93-20, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, October 1993.

[185] P. A. Lindsay. On transferring VDM verification techniques to Z. In Naftalin et al. [213], pages 190–213.

Also available as Technical Report 94-10, Department of Computer Science, University of Queensland, 1994.

[186] P. A. Lindsay and E. van Keulen. Case studies in the verification of specifications in VDM and Z. Technical Report 94-3, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, March 1994.

[187] M. Love. Animating Z specifications in SQL*Forms3.0. In Bowen and Nicholls [44], pages 294–306.

[188] P. J. Lupton. Promoting forward simulation. In Nicholls [224], pages 27–49.

[189] R. Macdonald. Z usage and abusage. Report no. 91003, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, February 1991.

This paper presents a miscellany of observations drawn from experience of using Z, shows a variety of techniques for expressing certain class of idea concisely and clearly, and alerts the reader to certain pitfalls which may trap the unwary.

[190] B. P. Mahony and I. J. Hayes. A case-study in timed refinement: A central heater. In Morris and Shaw [212], pages 138–149.

[191] B. P. Mahony and I. J. Hayes. A case-study in timed refinement: A mine pump. *IEEE Transactions on Software Engineering*, 18(9):817–826, September 1992.

[192] B. P. Mahony, C. Millerchip, and I. J. Hayes. A boiler control system: A case-study in timed refinement. Technical report, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 23 June 1993.

A specification and top-level design of a steam generating boiler system is presented as an example of the formal development of a real-time system.

[193] A. Martin. Encoding W: A logic for Z in 2OBJ. In Woodcock and Larsen [316], pages 462–481.

[194] P. Mataga and P. Zave. Formal specification of telephone features. In Bowen and Hall [38], pages 29–50.

[195] I. Maung and J. R. Howse. Introducing Hyper-Z – a new approach to object orientation in Z. In Bowen and Nicholls [44], pages 149–165.

[196] M. D. May. Use of formal methods by a silicon manufacturer. In C. A. R. Hoare, editor, *Developments in Concurrency and Communication*, University of Texas at Austin Year of Programming Series, chapter 4, pages 107–129. Addison-Wesley Publishing Company, 1990.

[197] M. D. May, G. Barrett, and D. E. Shepherd. Designing chips that work. In C. A. R. Hoare and M. J. C. Gordon, editors, *Mechanized Reasoning and Hardware Design*, pages 3–19. Prentice Hall International Series in Computer Science, 1992.

[198] M. D. May and D. E. Shepherd. Verification of the IMS T800 microprocessor. In *Proc. Electronic Design Automation*, pages 605–615, London, UK, September 1987.

[199] J. A. McDermid. Special section on Z. *IEE/BCS Software Engineering Journal*, 4(1):25–72, January 1989.

A special issue on Z, introduced and edited by Prof. J. A. McDermid. See also [32, 63, 269, 311].

[200] J. A. McDermid, editor. *The Theory and Practice of Refinement: Approaches to the Formal Development of Large-Scale Software Systems*. Butterworth Scientific, 1989.

This book contains papers from the 1st Refinement Workshop held at the University of York, UK, 7–8 January 1988. Z-related papers include [218, 171].

[201] J. A. McDermid. Formal methods: Use and relevance for the development of safety critical systems. In P. A. Bennett, editor, *Safety Aspects of Computer Control*. Butterworth-Heinemann, Oxford, UK, 1993.

[202] M. A. McMorran and S. Powell. *Z Guide for Beginners*. Blackwell Scientific, 1993.

[203] S. L. Meira and A. L. C. Cavalcanti. Modular object-oriented Z specifications. In Nicholls [224], pages 173–192.

[204] B. Meyer. On formalism in specifications. *IEEE Software*, 2(1):6–26, January 1985.

[205] J. D. Moffett and M. S. Sloman. A case study representing a model: To Z or not to Z? In Nicholls [224], pages 254–268.

[206] B. Q. Monahan and R. C. Shaw. Model-based specifications. In J. A. McDermid, editor, *Software Engineer's Reference Book*, chapter 21. Butterworth-Heinemann, Oxford, UK, 1991.

This chapter contains a case study in Z, followed by a discussion of the respective trade-offs in specification between Z and VDM.

[207] C. C. Morgan. *Programming from Specifications*. Prentice Hall International Series in Computer Science, 2nd edition, 1994.

This book presents a rigorous treatment of most elementary program development techniques, including iteration, recursion, procedures, parameters, modules and data refinement.

[208] C. C. Morgan and J. W. Sanders. Laws of the logical calculi. Technical Monograph PRG-78, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, September 1989.

This document records some important laws of classical predicate logic. It is designed as a reservoir to be tapped by *users* of logic, in system development.

[209] C. C. Morgan and B. A. Sufrin. Specification of the Unix filing system. *IEEE Transactions on Software Engineering*, 10(2):128–142, March 1984.

[210] C. C. Morgan and T. Vickers, editors. *On the Refinement Calculus*. Formal Approaches to Computing and Information Technology series (FACIT). Springer-Verlag, 1994.

This book collects together the work accomplished at Oxford on the refinement calculus: the rigorous development, from state-based assertional specification, of executable imperative code.

[211] C. C. Morgan and J. C. P. Woodcock, editors. *3rd Refinement Workshop*, Workshops in Computing. BCS-FACS, Springer-Verlag, 1991.

The workshop was held at the IBM Laboratories, Hursley Park, UK, 9–11 January 1990. See [258].

[212] J. M. Morris and R. C. Shaw, editors. *4th Refinement Workshop*, Workshops in Computing. BCS-FACS, Springer-Verlag, 1991.

The workshop was held at Cambridge, UK, 9–11 January 1991. For Z related papers, see [16, 155, 190, 307, 312, 304].

[213] M. Naftalin, T. Denvir, and M. Bertran, editors. *FME'94: Industrial Benefit of Formal Methods*, volume 873 of *Lecture Notes in Computer Science*. Formal Methods Europe, Springer-Verlag, 1994.

The 2nd FME Symposium was held at Barcelona, Spain, 24–28 October 1994. Z-related papers include [41, 69, 78, 101, 103, 108, 151, 185]. B-related papers include [80, 249, 285].

[214] K. T. Narayana and S. Dharap. Formal specification of a look manager. *IEEE Transactions on Software Engineering*, 16(9):1089–1103, September 1990.

A formal specification of the look manager of a dialog system is presented in Z. This deals with the presentation of visual aspects of objects and the editing of those visual aspects.

[215] K. T. Narayana and S. Dharap. Invariant properties in a dialog system. *ACM SIGSOFT Software Engineering Notes*, 15(4):67–79, September 1990.

[216] T. C. Nash. Using Z to describe large systems. In Nicholls [222], pages 150–178.

[217] Ph. W. Nehlig and D. A. Duce. GKS-9x: The design output primitive, an approach to specification. *Computer Graphics Forum*, 13(3):C–381–C–392, 1994.

[218] D. S. Neilson. Hierarchical refinement of a Z specification. In McDermid [200].

[219] D. S. Neilson. From Z to C: Illustration of a rigorous development method. Technical Monograph PRG-101, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, UK, 1990.

[220] D. S. Neilson. Machine support for Z: The zedB tool. In Nicholls [224], pages 105–128.

[221] D. S. Neilson and D. Prasad. zedB: A proof tool for Z built on B. In Nicholls [226], pages 243–258.

[222] J. E. Nicholls, editor. *Z User Workshop, Oxford 1989*, Workshops in Computing. Springer-Verlag, 1990.

Proceedings of the Fourth Annual Z User Meeting, Wolfson College & Rewley House, Oxford, UK, 14–15 December 1989. Published in collaboration with the British Computer Society. For the opening address see [233]. For individual papers, see [24, 54, 55, 68, 84, 110, 122, 134, 145, 158, 178, 216, 235, 262, 274, 302].

[223] J. E. Nicholls. A survey of Z courses in the UK. In *Z User Workshop, Oxford 1990* [224], pages 343–350.

[224] J. E. Nicholls, editor. *Z User Workshop, Oxford 1990*, Workshops in Computing. Springer-Verlag, 1991.

Proceedings of the Fifth Annual Z User Meeting, Lady Margaret Hall, Oxford, UK, 17–18 December 1990. Published in collaboration with the British Computer Society. For individual papers, see [17, 50, 57, 65, 112, 123, 136, 146, 148, 159, 162, 176, 203, 205, 220, 223, 230, 245, 256, 303, 320]. The proceedings also includes an *Introduction and Opening Remarks*, a *Selected Z Bibliography*, a selection of posters and information on Z tools.

[225] J. E. Nicholls. Domains of application for formal methods. In *Z User Workshop, York 1991* [226], pages 145–156.

[226] J. E. Nicholls, editor. *Z User Workshop, York 1991*, Workshops in Computing. Springer-Verlag, 1992.

Proceedings of the Sixth Annual Z User Meeting, York, UK. Published in collaboration with the British Computer Society. For individual papers, see [13, 19, 79, 58, 85, 92, 130, 221, 225, 239, 251, 263, 291, 296, 314, 324].

[227] J. E. Nicholls. Plain guide to the Z base standard. In Bowen and Nicholls [44], pages 52–61.

[228] C. J. Nix and B. P. Collins. The use of software engineering, including the Z notation, in the development of CICS. *Quality Assurance*, 14(3):103–110, September 1988.

[229] A. Norcliffe and G. Slater. *Mathematics for Software Construction*. Series in Mathematics and its Applications. Ellis Horwood, 1991.

Contents: Why mathematics; Getting started: sets and logic; Developing ideas: schemas; Functions; Functions in action; A real problem from start to finish: a drinks machine; Sequences; Relations; Generating programs from specifications: refinement; The role of proof; More examples of specifications; Concluding remarks; Answers to exercises.

[230] A. Norcliffe and S. H. Valentine. A video-based training course in reading Z specifications. In Nicholls [224], pages 337–342.

[231] Z readers video course. PAVIC Publications, 1992. Sheffield Hallam University, 33 Collegiate Crescent, Sheffield S10 2BP, UK.

Video-based Training Course on the Z Specification Language. The course consists of 5 videos, each of approximately one hour duration, together with supporting texts and case studies.

[232] G. Normington. Cleanroom and Z. In Bowen and Nicholls [44], pages 281–293.

[233] B. Oakley. The state of use of formal methods. In Nicholls [222], pages 1–5.

A record of the opening address at ZUM'89.

[234] C. E. Parker. Z tools catalogue. ZIP project report ZIP/BAe/90/020, British Aerospace, Software Technology Department, Warton PR4 1AX, UK, May 1991.

[235] M. Phillips. CICS/ESA 3.1 experiences. In Nicholls [222], pages 179–185.

Z was used to specify 37,000 lines out of 268,000 lines of code in the IBM CICS/ESA 3.1 release. The initial development benefit from using Z was assessed as being a 9% improvement in the *total development cost* of the release, based on the reduction of programmer days fixing problems.

[236] M. Pilling, A. Burns, and K. Raymond. Formal specifications and proofs of inheritance protocols for real-time scheduling. *IEE/BCS Software Engineering Journal*, 5(5):263–279, September 1990.

[237] P. R. H. Place and K. C. Kang. Safety-critical software: Status report and annotated bibliography. Technical Report CMU/SEI-92-TR-5 & ESC-TR-93-182, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, USA, June 1993.

[238] F. Polack and K. C. Mander. Software quality assurance using the SAZ method. In Bowen and Hall [38], pages 230–249.

[239] F. Polack, M. Whiston, and P. Hitchcock. Structured analysis – a draft method for writing Z specifications. In Nicholls [226], pages 261–286.

[240] F. Polack, M. Whiston, and K. Mander. The SAZ project: Integrating SSADM and Z. In Woodcock and Larsen [316], pages 541–557.

[241] B. F. Potter, J. E. Sinclair, and D. Till. *An Introduction to Formal Specification and Z*. Prentice Hall International Series in Computer Science, 1991.

Contents: Formal specification in the context of software engineering; An informal introduction to logic and set theory; A first specification; The Z notation: the mathematical language, relations and functions, schemas and specification structure; A first specification revisited; Formal reasoning; From specification to program: data and operation refinement, operation decomposition; From theory to practice.

[242] S. Prehn and W. J. Toetenel, editors. *VDM'91: Formal Software Development Methods*, volume 551 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991. Volume 1: Conference Contributions.

The 4th VDM-Europe Symposium was held at Noordwijkerhout, The Netherlands, 21–25 October 1991. Papers with relevance to Z include [12, 25, 74, 89, 115, 147, 299, 306, 323]. See also [243].

[243] S. Prehn and W. J. Toetenel, editors. *VDM'91: Formal Software Development Methods*, volume 552 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991. Volume 2: Tutorials.

Papers with relevance to Z include [4, 313]. See also [242].

[244] G-H. B. Rafsanjani and S. J. Colwill. From Object-Z to C^{++}: A structural mapping. In Bowen and Nicholls [44], pages 166–179.

[245] G. P. Randell. Data flow diagrams and Z. In Nicholls [224], pages 216–227.

[246] G. P. Randell. Improving the translation from data flow diagrams into Z by incorporating the data dictionary. Report no. 92004, RSRE, Ministry of Defence, Malvern, Worcestershire, UK, January 1992.

[247] B. Ratcliff. *Introducing Specification Using Z: A Practical Case Study Approach*. International Series in Software Engineering. McGraw-Hill, 1994.

[248] N. R. Reizer, G. D. Abowd, B. C. Meyers, and P. R. H. Place. Using formal methods for requirements specification of a proposed POSIX standard. In *IEEE International Conference on Requirements Engineering (ICRE'94)*, April 1994.

[249] B. Ritchie, J. Bicarregui, and H. P. Haughton. Experiences in using the abstract machine notation in a GKS case study. In Naftalin et al. [213], pages 93–104.

[250] A. R. Ruddle. Formal methods in the specification of real-time, safety-critical control systems. In Bowen and Nicholls [44], pages 131–146.

[251] M. Saaltink. Z and Eves. In Nicholls [226], pages 223–242.

[252] H. Saiedian. The mathematics of computing. *Journal of Computer Science Education*, 3(3):203–221, 1992.

[253] A. C. A. Sampaio and S. L. Meira. Modular extensions to Z. In Bjørner et al. [27], pages 211–232.

[254] S. A. Schuman and D. H. Pitt. Object-oriented subsystem specification. In L. G. L. T. Meertens, editor, *Program Specification and Transformation*, pages 313–341. Elsevier Science Publishers (North-Holland), 1987.

[255] S. A. Schuman, D. H. Pitt, and P. J. Byers. Object-oriented process specification. In C. Rattray, editor, *Specification and Verification of Concurrent Systems*, Workshops in Computing, pages 21–70. Springer-Verlag, 1990.

[256] L. T. Semmens and P. M. Allen. Using Yourdon and Z: An approach to formal specification. In Nicholls [224], pages 228–253.

[257] L. T. Semmens, R. B. France, and T. W. G. Docker. Integrated structured analysis and formal specification techniques. *The Computer Journal*, 35(6):600–610, December 1992.

[258] C. T. Sennett. Using refinement to convince: Lessons learned from a case study. In Morgan and Woodcock [211], pages 172–197.

[259] C. T. Sennett. Demonstrating the compliance of Ada programs with Z specifications. In Jones et al. [160].

[260] D. E. Shepherd. Verified microcode design. *Microprocessors and Microsystems*, 14(10):623–630, December 1990.

This article is part of a special issue on *Formal aspects of microprocessor design*, edited by H. S. M. Zedan. See also [33].

[261] D. E. Shepherd and G. Wilson. Making chips that work. *New Scientist*, 1664:61–64, May 1989.

A general article containing information on the formal development of the T800 floating-point unit for the transputer including the use of Z.

[262] A. Smith. The Knuth-Bendix completion algorithm and its specification in Z. In Nicholls [222], pages 195–220.

[263] A. Smith. On recursive free types in Z. In Nicholls [226], pages 3–39.

[264] G. Smith. *An Object-Oriented Approach to Formal Specification*. PhD thesis, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, October 1992.

A detailed description of a version of Object-Z similar to (but not identical to) that in [97]. The thesis also includes a formalization of temporal logic history invariants and a fully-abstract model of classes in Object-Z.

[265] G. Smith. A object-oriented development framework for Z. In Bowen and Hall [38], pages 89–107.

[266] G. Smith and R. Duke. Modelling a cache coherence protocol using Object-Z. In *Proc. 13th Australian Computer Science Conference (ACSC-13)*, pages 352–361, 1990.

[267] P. Smith and R. Keighley. The formal development of a secure transaction mechanism. In Prehn and Toetenel [242], pages 457–476. Volume 1: Conference Contributions.

[268] J. M. Spivey. *Understanding Z: A Specification Language and its Formal Semantics*, volume 3 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, January 1988.

Published version of 1985 DPhil thesis.

[269] J. M. Spivey. An introduction to Z and formal specifications. *IEE/BCS Software Engineering Journal*, 4(1):40–50, January 1989.

[270] J. M. Spivey. A guide to the `zed` style option. Oxford University Computing Laboratory, December 1990.

A description of the Z style option '`zed.sty`' for use with the LaTeX document preparation system [174].

[271] J. M. Spivey. Specifying a real-time kernel. *IEEE Software*, 7(5):21–28, September 1990.

This case study of an embedded real-time kernel shows that mathematical techniques have an important role to play in documenting systems and avoiding design flaws.

[272] J. M. Spivey. *The fUZZ Manual*. Computing Science Consultancy, 34 Westlands Grove, Stockton Lane, York YO3 0EF, UK, 2nd edition, July 1992.

The manual describes a Z type-checker and '`fuzz.sty`' style option for LaTeX documents [174]. The package is compatible with the book, *The Z Notation: A Reference Manual* by the same author [273].

[273] J. M. Spivey. *The Z Notation: A Reference Manual*. Prentice Hall International Series in Computer Science, 2nd edition, 1992.

This is a revised edition of the first widely available reference manual on Z originally published in 1989. The book provides a complete and definitive guide to the use of Z in specifying information systems, writing specifications and designing implementations. See also the draft Z standard [52].

Contents: Tutorial introduction; Background; The Z language; The mathematical tool-kit; Sequential systems; Syntax summary; Changes from the first edition; Glossary.

[274] J. M. Spivey and B. A. Sufrin. Type inference in Z. In Nicholls [222], pages 6–31.

Also published as [275].

[275] J. M. Spivey and B. A. Sufrin. Type inference in Z. In Bjørner et al. [27], pages 426–438.

[276] S. Stepney. *High Integrity Compilation: A Case Study*. Prentice Hall, 1993.

[277] S. Stepney and R. Barden. Annotated Z bibliography. *Bulletin of the European Association of Theoretical Computer Science*, 50:280–313, June 1993.

[278] S. Stepney, R. Barden, and D. Cooper, editors. *Object Orientation in Z*. Workshops in Computing. Springer-Verlag, 1992.

This is a collection of papers describing various OOZ approaches – Hall, ZERO, MooZ, Object-Z, OOZE, Schuman & Pitt, Z^{++}, ZEST and Fresco (an object-oriented VDM method) – in the main written by the methods' inventors, and all specifying the same two examples. The collection is a revised and expanded version of a ZIP report distributed at the 1991 Z User Meeting at York.

[279] S. Stepney, R. Barden, and D. Cooper. A survey of object orientation in Z. *IEE/BCS Software Engineering Journal*, 7(2):150–160, March 1992.

[280] S. Stepney and S. P. Lord. Formal specification of an access control system. *Software – Practice and Experience*, 17(9):575–593, September 1987.

[281] P. Stocks. *Applying formal methods to software testing*. PhD thesis, Department of Computer Science, University of Queensland, St. Lucia 4072, Australia, 1993.

[282] P. Stocks and D. A. Carrington. Deriving software test cases from formal specifications. In *6th Australian Software Engineering Conference*, pages 327–340, July 1991.

[283] P. Stocks and D. A. Carrington. Test template framework: A specification-based testing case study. In *Proc. International Symposium on Software Testing and Analysis (ISSTA'93)*, pages 11–18, June 1993.

Also available in a longer form as Technical Report UQCS-255, Department of Computer Science, University of Queensland.

[284] P. Stocks and D. A. Carrington. Test templates: A specification-based testing framework. In *Proc. 15th International Conference on Software Engineering*, pages 405–414, May 1993.

Also available in a longer form as Technical Report UQCS-243, Department of Computer Science, University of Queensland.

[285] A. C. Storey and H. P. Haughton. A strategy for the production of verifiable code using the B method. In Naftalin et al. [213], pages 346–365.

[286] B. A. Sufrin. Formal methods and the design of effective user interfaces. In M. D. Harrison and A. F. Monk, editors, *People and Computers: Designing for Usability*. Cambridge University Press, 1986.

[287] B. A. Sufrin. Formal specification of a display-oriented editor. In N. Gehani and A. D. McGettrick, editors, *Software Specification Techniques*, International Computer Science Series, pages 223–267. Addison-Wesley Publishing Company, 1986.

Originally published in Science of Computer Programming, 1:157–202, 1982.

[288] B. A. Sufrin. Effective industrial application of formal methods. In G. X. Ritter, editor, *Information Processing 89, Proc. 11th IFIP Computer Congress*, pages 61–69. Elsevier Science Publishers (North-Holland), 1989.

This paper presents a Z model of the Unix *make* utility.

[289] B. A. Sufrin and He Jifeng. Specification, analysis and refinement of interactive processes. In M. D. Harrison and H. Thimbleby, editors, *Formal Methods in Human-Computer Interaction*, volume 2 of *Cambridge Series on Human-Computer Interaction*, chapter 6, pages 153–200. Cambridge University Press, 1990.

A case study on using Z for process modelling.

[290] P. A. Swatman. Using formal specification in the acquisition of information systems: Educating information systems professionals. In Bowen and Nicholls [44], pages 205–239.

[291] P. A. Swatman, D. Fowler, and C. Y. M. Gan. Extending the useful application domain for formal methods. In Nicholls [226], pages 125–144.

[292] D. Till, editor. *6th Refinement Workshop*, Workshop in Computing. BCS-FACS, Springer-Verlag, 1994.

The workshop was held at City University, London, UK, 5–7 January 1994. See [109, 177].

[293] D. Till and B. F. Potter. The specification in Z of gateway functions within a communications network. In *Proc. IFIP WG10.3 Conference on Distributed Processing*. Elsevier Science Publishers (North-Holland), October 1987.

[294] R. Took. The presenter – a formal design for an autonomous display manager. In I. Sommerville, editor, *Software Engineering Environments*, pages 151–169. Peter Peregrinus, London, 1986.

[295] I. Toyn and J. A. McDermid. CADiZ: An architecture for Z tools and its implementation. Technical document, Computer Science Department, University of York, York YO1 5DD, UK, November 1993.

[296] S. H. Valentine. Z^{--}, an executable subset of Z. In Nicholls [226], pages 157–187.

[297] S. H. Valentine. Putting numbers into the mathematical toolkit. In Bowen and Nicholls [44], pages 9–36.

[298] M. J. van Diepen and K. M. van Hee. A formal semantics for Z and the link between Z and the relational algebra. In Bjørner et al. [27], pages 526–551.

[299] K. M. van Hee, L. J. Somers, and M. Voorhoeve. Z and high level Petri nets. In Prehn and Toetenel [242], pages 204–219. Volume 1: Conference Contributions.

[300] M. M. West and B. M. Eaglestone. Software development: Two approaches to animation of Z specifications using Prolog. *IEE/BCS Software Engineering Journal*, 7(4):264–276, July 1992.

[301] C. Wezeman and A. Judge. Z for managed objects. In Bowen and Hall [38], pages 108–119.

[302] R. W. Whitty. Structural metrics for Z specifications. In Nicholls [222], pages 186–191.

[303] P. J. Whysall and J. A. McDermid. An approach to object-oriented specification using Z. In Nicholls [224], pages 193–215.

[304] P. J. Whysall and J. A. McDermid. Object-oriented specification and refinement. In Morris and Shaw [212], pages 151–184.

[305] J. M. Wing. A specifier's introduction to formal methods. *IEEE Computer*, 23(9):8–24, September 1990.

[306] J. M. Wing and A. M. Zaremski. Unintrusive ways to integrate formal specifications in practice. In Prehn and Toetenel [242], pages 545–570. Volume 1: Conference Contributions.

[307] K. R. Wood. The elusive software refinery: a case study in program development. In Morris and Shaw [212], pages 281–325.

[308] K. R. Wood. A practical approach to software engineering using Z and the refinement calculus. *ACM Software Engineering Notes*, 18(5):79–88, December 1993.

[309] J. C. P. Woodcock. Calculating properties of Z specifications. *ACM SIGSOFT Software Engineering Notes*, 14(4):43–54, 1989.

[310] J. C. P. Woodcock. Mathematics as a management tool: Proof rules for promotion. In *Proc. 6th Annual CSR Conference on Large Software Systems*, Bristol, UK, September 1989.

[311] J. C. P. Woodcock. Structuring specifications in Z. *IEE/BCS Software Engineering Journal*, 4(1):51–66, January 1989.

[312] J. C. P. Woodcock. Implementing promoted operations in Z. In Morris and Shaw [212], pages 366–378.

[313] J. C. P. Woodcock. A tutorial on the refinement calculus. In Prehn and Toetenel [243], pages 79–140. Volume 2: Tutorials.

[314] J. C. P. Woodcock and S. M. Brien. W: A logic for Z. In Nicholls [226], pages 77–96.

[315] J. C. P. Woodcock, P. H. B. Gardiner, and J. R. Hulance. The formal specification in Z of Defence Standard 00-56. In Bowen and Hall [38], pages 9–28.

[316] J. C. P. Woodcock and P. G. Larsen, editors. *FME'93: Industrial-Strength Formal Methods*, volume 670 of *Lecture Notes in Computer Science*. Formal Methods Europe, Springer-Verlag, 1993.

The 1st FME Symposium was held at Odense, Denmark, 19–23 April 1993. Z-related papers include [45, 72, 107, 154, 193, 240].

[317] J. C. P. Woodcock and M. Loomes. *Software Engineering Mathematics: Formal Methods Demystified*. Pitman, 1988.

[318] J. C. P. Woodcock and C. C. Morgan. Refinement of state-based concurrent systems. In Bjørner et al. [27], pages 340–351.

Work on combining Z and CSP.

[319] R. Worden. Fermenting and distilling. In Bowen and Hall [38], pages 1–6.

[320] J. B. Wordsworth. The CICS application programming interface definition. In Nicholls [224], pages 285–294.

[321] J. B. Wordsworth. *Software Development with Z*. Addison-Wesley, 1992.

This book provides a guide to developing software from specification to code, and is based in part on work done at IBM's UK Laboratory that won the UK Queen's Award for Technological Achievement in 1992.

Contents: Introduction; A simple Z specification; Sets and predicates; Relations and functions; Schemas and specifications; Data design; Algorithm design; Specification of an oil terminal control system.

[322] Xiaoping Jia. *ZTC: A Type Checker for Z – User's Guide*. Institute for Software Engineering, Department of Computer Science and Information Systems, DePaul University, Chicago, IL 60604, USA, 1994.

ZTC is a type checker for the Z specification language. ZTC accepts two forms of input: LaTeX with `zed` style option and ZSL, an ASCII version of Z. ZTC can also perform translations between the two input forms. This document is intended to serve as both a user's guide and a reference manual for ZTC.

[323] P. Zave and M. Jackson. Techniques for partial specification and specification of switching systems. In Prehn and Toetenel [242], pages 511–525. Volume 1: Conference Contributions.

Also published as [324].

[324] P. Zave and M. Jackson. Techniques for partial specification and specification of switching systems. In Nicholls [226], pages 205–219.

[325] Y. Zhang and P. Hitchcock. EMS: Case study in methodology for designing knowledge-based systems and information systems. *Information and Software Technology*, 33(7):518–526, September 1991.